

Solving Multi-objective Optimisation Problems Using the Potential Pareto Regions Evolutionary Algorithm

Nasreddine Hallam, Graham Kendall, and Peter Blanchfield

School of Computer Science and IT, The University of Nottingham

Abstract. In this paper we propose a novel multi-objective evolutionary algorithm that we call Potential Pareto Regions Evolutionary Algorithm (PPREA). Unlike state-of-the-art algorithms, which use a fitness assignment method based on Pareto ranking, the approach adopted in this work is new. The fitness of an individual is equal to the least improvement needed by that individual in order to reach non-dominance status.

This new algorithm is compared against the Nondominated Sorting Genetic Algorithm (NSGA-II) on a set of test suite problems derived from the works of researchers from MOEA community.

1 Introduction

Many specialised algorithms have been devised to tackle multi-objective problems. In recent years, a new wave of algorithms has emerged as the major ‘stakeholder’ in this field of research. The most attractive feature of these algorithms is their capability to search for a set of solutions in a single run by considering a population of potential solutions. These are Multi-Objective Evolutionary Algorithms (MOEAs).

Recently, elitist-MOEAs (archive-based MOEAs) have dominated the field of MOEAs. Essentially, they are the standard MOEAs supplied with an elitist archive as proposed by (Zitzler and Thiele 1999; Knowles and Corne 1999). These algorithms have proven to be successful in solving complex real world problems (Burke et al 2006).

In this work a novel MOEA, called Potential Pareto Regions Evolutionary Algorithm (PPREA), is formulated and developed. This algorithm is based on the new approach of Potential Pareto Regions (PPRs) (Hallam 2005).

This paper is organised as follows. Section 2 briefly describes the related work. NSGA-II being the most popular MOEA is briefly presented. Section 3 describes and proposes the novel MOEA. Section 4 is an empirical study of the PPREA. The latter is tested against NSGA-II on a set of suitably chosen test problems. Lastly, concluding remarks are given in section 5.

2 Related Work

The literature is rich of successful MOEAs. Strength Pareto Evolutionary Algorithm (SPEA2) of (Zitzler et al. 2001) and Nondominated Sorting Genetic

Algorithm (NSGA-II) of (Deb et al. 2001) are the most widely used. In our comparative study, we choose NSGA-II as it is simple to implement. Therefore, this algorithm is briefly explained in this section.

Following criticisms (see Knowles and Corne 1999 ; Zitzler and Thiele 1999), a new improved NSGA, coined NSGA-II, has been designed and tested by (Deb et al. 2001) specifically to address those criticisms.

The improvements brought to NSGA were mainly concerning three basic points:

elitism NSGA-II implements an elitist strategy to improve convergence.

speed NSGA-II employs a fast nondominance sorting approach based on a better book-keeping technique.

diversity NSGA-II uses a density preservation technique that is better than the niching and fitness sharing technique.

3 Potential Pareto Regions Evolutionary Algorithm (PPREA): A New MOEA

The theoretical foundation of the proposed algorithm is briefly described. Potential Pareto Regions Evolutionary Algorithm (PPREA) is designed based on a new fitness assignment scheme and a new diversity preservation technique. These two are detailed in (Hallam 2005, Hallam et al. 2005).

3.1 Definitions and Key Concepts (Hallam 2005)

Definition 1. A PPR is hyper-area delimited by two immediate neighbouring nondominated points of the archive.

These PPRs are dynamic regions in the objective space within which any generated vector solution is automatically nondominated with regard to all the current solutions of the archive.

Let A_t be the archive of the current population and *MaxArchiveSize* its maximum size.

Let $z_t = Gen(t)$ be a solution generated by the function *Gen* at iteration t .

Let PPR^{xy} be a PPR delimited by two (neighbouring) nondominated solutions x and y of A_t . For the proof of the following lemmas, one can refer to (Hallam 2005).

Lemma 1. If there exist a PPR which contains z_t , then z_t is incomparable to any point in the archive ($z_t \in PPR^{xy} \Rightarrow \forall a \in A_t, z_t \sim a$).

Then, constructing the chain of PPRs is done by determining the set of PPRs. For this purpose, the $z^i, i = 1, |A_t| - 1 \wedge z^i \in A_t$ are arranged in a way that $z_{i,k} > z_{i+1,k}$ for a given k . Without loss of generality we set k to 1.

Definition 2. *SPPR* is the set of the PPRs ordered according to ‘neighbourhood’ order. $SPPR = \{PPR^{z_i z_{i+1}} \mid z_i, z_{i+1} \in A_t \wedge z_{i,k} > z_{i+1,k}, i = 1..|A_t| - 1\}$

After determining the set $SPPR$, the task is to find a closest PPR to each point in the objective space. This is done thanks to the following lemma (depicted in Fig. 1). This also forms the basis of the fitness assignment scheme. Let $Dom(A_t, g)$ be the set of all points in the archive dominating g ($g \notin A_t$). Let us define the set of those PPRs dominating a point b .

Definition 3. $DomPPR(g)$ is the set of those PPRs whose at least one of their end-points is in $Dom(A_t, g)$. $DomPPR(b) = \{R^{xy} \mid x \in Dom(A_t, g) \vee y \in Dom(A_t, g)\}$.

Lemma 2. a. The minimum distance of a point g to $SPPR$ is the minimum distance to one of the PPRs in $DomPPR(g)$.

b. For every two consecutive points in $Dom(A_t, g)$, the minimum distance from the point g to their PPR is from the point to the upper right corner of that region.

c. If the $Dom(A_t, g)$ is a singleton $\{d\}$, then the minimum two distances from the point b to the two PPRs of d , (PPR^{-d} and PPR^{d-}), are respectively $(g_1 - d_1)$ and $(g_2 - d_2)$.

3.2 PPREA: The Algorithm

MOEAs differ only in the fitness assignment scheme and the archive update process. Sampling, crossover, mutation, and re-insertion are basically the same. It has been shown in (Burke and Silva 2006) that the choice of a fitness evaluation method does have an influence on the performance of a multi-objective optimiser. On the other hand, diversity is as important as the fitness assignment scheme not only for diversifying the solutions, but also on guiding the search process (see Silva and Burke 2004).

Algorithm 1 PPREA: The algorithm - (Hallam 2005)

Input: N (Population size),

MaxArchiveSize, M (maximum number of generations)

Ouput: A (The archive)

Step 0: Initialisation

generate an initial Population Pop_0 , Set $t = 0$ and

create the empty archive (external set) $A_t = \emptyset$

Step 1: FITNESS ASSIGNMENT: Calculate the fitness values of all individuals in Pop_t

Step 2: $A_{t+1} = \text{ARCHIVE UPDATE}(A_t)$

Step 3: Termination.

if $t > M$ or another stopping condition is met **then**

$A = \text{filterFront}(A_{t+1})$ and Stop.

end if

Step 4: Mating Selection (tournament selection)

Step 5: $Q_{t+1} = \text{Variation}(A_{t+1})$: Produce offspring using Crossover and Mutation

$Pop_{t+1} = A_{t+1} \cup Q_{t+1}$, $t = t + 1$ and GO TO Step 1.

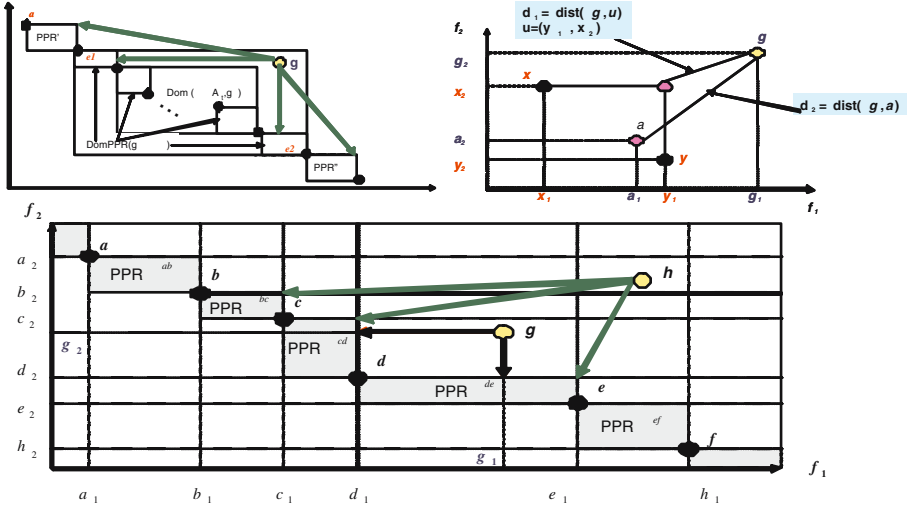


Fig. 1. Illustration of Lemma 2

Fitness Assignment. Minimisation is adopted in PPREA. Individuals of the archive are assigned negative values, while the remaining individuals are assigned positive values. The nondominated extremums of the archive are assigned a fitness value that is (in absolute value) twice the size of the largest PPR, whereas the rest of the archive members are assigned values equal (in absolute value) to the sum of the sizes of the two adjacent PPRs.

The dominated individuals are assigned fitness values equal to their respective Euclidean distances to the nearest PPR as calculated in lemmas 2. This is called the Expected Improvement which, besides being the cornerstone of the fitness assignment scheme, is also used as the building block of a quality performance indicator (Hallam 2005).

Archive Update. Whenever the size of the current Pareto front is less than *MaxArchiveSize*, then the empty slots need to be populated with adequate individuals from the rest of the population. These individuals should be fit and not concentrated in one or few sub-spaces.

The rest of the population is re-arranged into a set of lists. Recall that non-dominated individuals are sorted according to the first objective. Then each of these individuals is assigned a list of those points that it dominates. These lists are sorted in an ascending order according to the distances to the PPR of the corresponding nondominated point (fitness of the dominated individuals). Then, the filling process is a cyclic removal of the best individuals from each list. While the archive is not full, we take from each list the first individual and insert it into the archive. If the archive is still not full, we take the second individual from each list and insert it into the archive, and so forth.

The other situation is when the size of the archive containing nondominated individuals is greater than *MaxArchiveSize*. The nondominated points in excess

and found in the most crowded regions must be removed. Thus, the removal process is iterative. First the smallest PPR is found. Thereafter, if one of the individuals of this PPR is an extremum, then the other individual is removed; otherwise, from the two individuals, we remove the one whose other adjacent PPR is the smallest.

4 Numerical Testing and Analysis

The PPREA is tested and compared with NSGA-II – one of the most successful MOEA in the literature. In other experimental comparative studies (Zitzler et al. 2001), SPEA2 has been shown to be as effective as NSGA-II. We choose (for the comparative study) NSGA-II as it is more efficient (fast in execution) and simple to implement.

The test functions used in the experiments reported herein are exactly those used by (Deb et al. 2001) when first they proposed NSGA-II. The popularity of their algorithm has in fact started after it has excelled in outperforming other MOEAs on these test problems. Due to space limit, the reader is referred to (Deb et al. 2001) for a complete and detailed listing of these test suite functions. Also, only the results of the ZDTs test functions are shown in this paper. Reference (Hallam 2005) contains an exhaustive empirical study of PPREA and its comparison with NSGA-II.

Table 1. ZDT1’s Test Results (100, 250, 500, and 1000 generations)

Gen	100		250	
MOEA	PPREA	NSGA-II	PPREA	NSGA-II
SizeFrontMOEA	124	78	544	350
ParetoSize	86		687	
InsidePareto	11	75	520	167
OutsidePareto	113	3	137	186
PercentFromPareto	12.79	87.20	75.69	24.30
PercentFromFRONT	8.87	96.15	95.58	47.71
AvergaeEI	0.022	0.0009	0.06	0.002
StdvEI	0.012	0.001	0.034	0.015
AverageTimeExec	0.7	0.6	1.7	1.6
Generation	500		1000	
MOEA	PPREA	NSGA-II	PPREA	NSGA-II
SizeFrontMOEA	938	964	910	908
ParetoSize	1847		1804	
InsidePareto	909	938	904	900
OutsidePareto	148	215	154	223
PercentFromPareto	49.21	50.78	50.11	49.88
PercentFromFRONT	96.90	97.30	99.34	99.11
AvergaeEI	0.048	0.003	0.054	0.002
StdvEI	0.038	0.017	0.038	0.005
AverageTimeExec	3.4	3.3	7	6.8

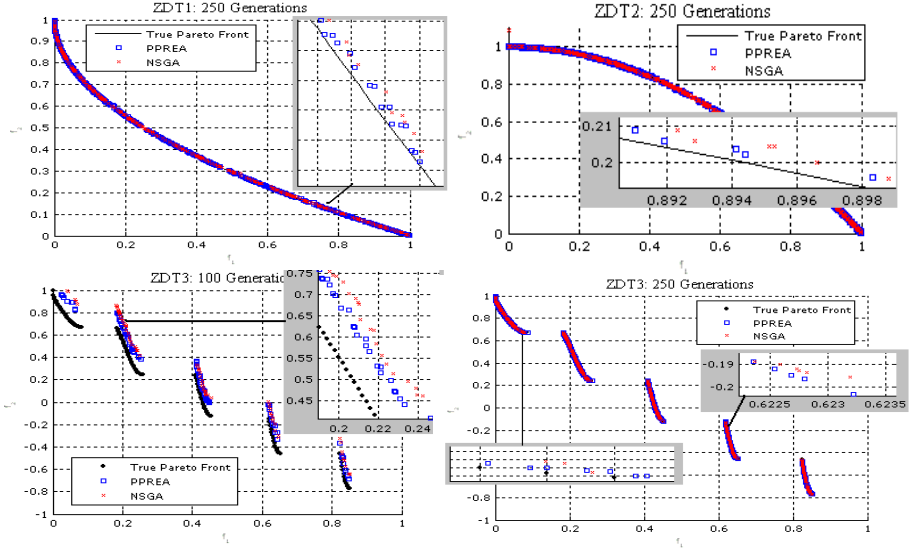


Fig. 2. Convergence Results for ZDT1, ZDT2 and ZDT3 Test Functions

Per algorithm and test function, the outcomes of ten runs were unified. A front is filtered from the unified set. In order to get a glimpse on the pace of the convergence, we used the following maximum number of generations (100, 250, 500, and 1000). Furthermore, each simulation run was carried out independently of the test function using the following parameters:

- Size of the population and the archive is 100
- Individual representation Real
- Simulated Binary Crossover (SBX) with probability 0.9 and a Polynomial Mutation probability $1/\text{numberOfVariables}$. The distribution indices for the SBX and the mutation are set to 20 ($\eta_m = \eta_c = 20$).

Measuring the Performance. A Common Pareto Front (CPF) is filtered from both algorithms: $CPF = ND(PPREA \cup NSGA - II)$. Two main performance values are then computed. The percentage of a front in the common archive (which is the error ratio), and the average of the cumulative Expected Improvement (see Hallam 2005) of its residual set with regards to SPPR. Also, the set of vector solutions of each MOEA that are in the CPF is also derived: $MOEA_{Pareto} = MOEA \cap CPF$. Let $MOEA \in \{PPREA, NSGA - II\}$

For each simulation run, the following results are reported:

- The sizes of the CPF and the two fronts returned by each MOEA.
- The number of vector solutions of each MOEA *inside* and *outside* the CPF.
- The % of each front covering the CPF: $\frac{|MOEA_{Pareto}|}{|CPF|}$.
- The % of each MOEA Pareto covering the front returned by the MOEA in question: $\frac{|MOEA_{Pareto}|}{|MOEA|}$.

Table 2. ZDT2’s Test Results (100, 250, 500, and 1000 generations) A = PPREA, B= NSGA-II

Table 4. ZDT4’s Test Results (100, 250, 500, and 1000 generations)

Generation	100		250	
MOEA	A	B	A	B
SizeMOEA	36	42	431	442
ParetoSize	41		679	
In Pareto	36	5	365	314
Out Pareto	0	37	66	165
%Pareto	87.80	12.19	53.75	46.24
%FRONT	100	11.90	84.68	71.04
AvergaeEI	0	0.088	0.0008	0.063
StdvEI	0	0.038	0.001	0.121
AvgTimExe	0.7	0.6	1.8	1.5

Table 3. ZDT3’s Test Results (100, 250, 500, and 1000 generations)

Generation	100		250	
MOEA	A	B	A	B
SizeMOEA	98	91	429	463
ParetoSize	101		677	
In Pareto	97	4	379	298
Out Pareto	1	87	51	252
% Pareto	96.03	3.96	55.98	44.01
% Front	98.97	4.39	88.34	64.36
AvergaeEI	0.0002	0.015	0.0008	0.020
StdvEI	0	0.023	0.001	0.040
AvgTimExe	0.7	0.6	1.8	1.6

Generation	100		250	
MOEA	A	B	A	B
SizeMOEA	94	98	283	403
ParetoSize	102		476	
In Pareto	94	8	253	223
Out Pareto	0	90	30	270
% Pareto	92.15	7.84	53.15	46.84
% Front	100	8.16	89.39	55.33
AvergaeEI	0	0.10	0.002	0.045
StdvEI	0	0.038	0.001	0.065
AvgTimExe	0.6	0.4	1.6	1.3
Generation	500		1000	
MOEA	A	B	A	B
SizeMOEA	654	514	759	603
ParetoSize	864		1129	
In Pareto	616	248	690	439
Out Pareto	144	446	213	610
% Pareto	71.29	28.70	61.11	38.88
% Front	94.18	48.24	90.90	72.80
AvergaeEI	0.002	0.024	0.001	0.02
StdvEI	0.002	0.052	0.002	0.047
AvgTimExe	3.3	2.7	6.8	5.7

Table 5. ZDT6’s Test Results (100, 250, 500, and 1000 generations)

Generation	100		250	
MOEA	A	B	A	B
SizeMOEA	97	104	669	721
ParetoSize	133		1140	
In Pareto	83	50	616	524
Out Pareto	14	54	67	251
% Pareto	62.40	37.59	54.03	45.96
% Front	85.56	48.07	92.07	72.67
AvergaeEI	0.003	0.004	0.005	0.007
StdvEI	0.002	0.003	0.015	0.012
AvgTimExe	0.6	0.5	1.6	1.4

- The average and standard deviation of the Expected Improvement (EI) of each MOEA.
- The average Execution Time of each MOEA.

ZDT1's Test Results. Referring to Table 1, NSGA-II is clearly outperforming PPREA at Generation 100 test (early stage). However, PPREA clearly outperforms NSGA-II at Generation 250 test (see the top-left figure in Fig. 2).

ZDT2's Test Results. Refer to Table 2, and the top-right figure in Fig. 2. At Generation 100 test, PPREA is almost totally outperforming NSGA-II. Notice also that the vector solutions of the front returned by PPREA are *all* in the

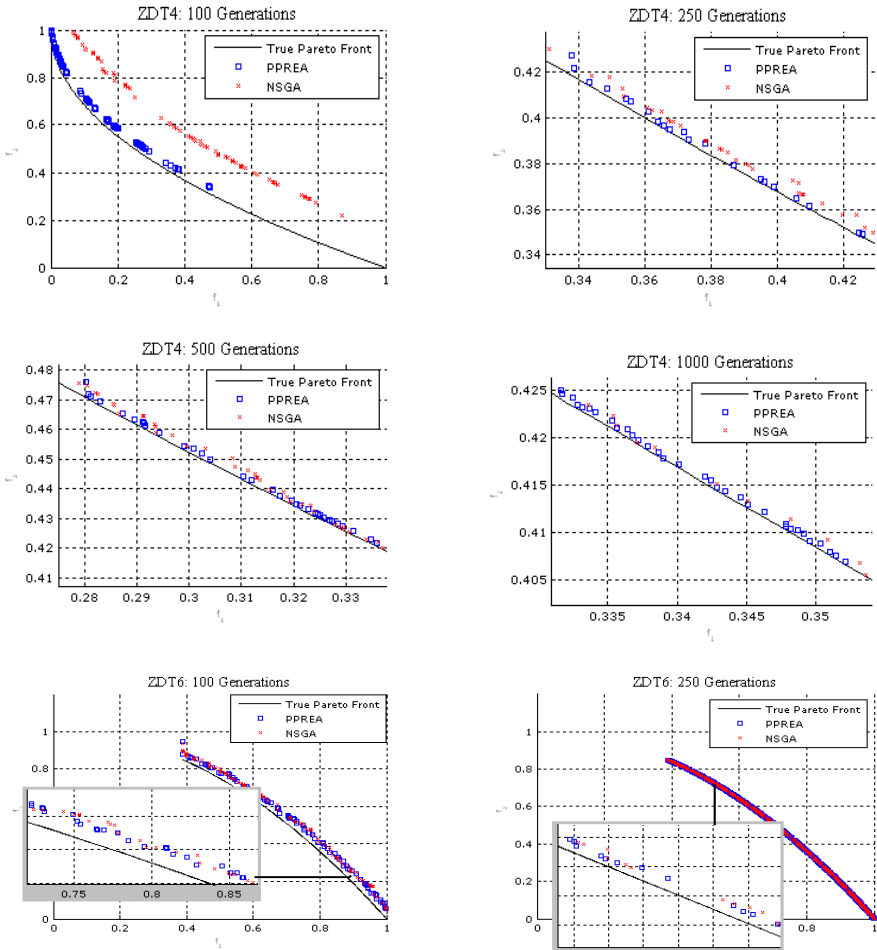


Fig. 3. Convergence Results for ZDT4 and ZDT6 Test Functions

CPF. In Generation 250 and Generation 1000 tests, PPREA is also clearly outperforming NSGA-II. In Generation 500 test, both algorithms seem to fairly share the CPF.

ZDT3's Test Results. One can see from Table 3 that PPREA is monopolising the CPF in the Generation 100 test. Only 1 vector solution returned by PPREA is outside the CPF (but still very close to it ($AverageEI = 0.0002$), compared to 87 vector solutions returned by NSGA-II out of 91 being outside the CPF (see the two bottom figures in Fig. 2). PPREA outperforms (to some extent) NSGA-II in Generation 250 test. In the remaining simulation tests, PPREA and NSGA-II fairly share the CPF. However, the vectors solutions returned by PPREA and left behind the CPF are much closer on average to the CPF than those of NSGA-II. See also the two bottom figures in Fig 2.

ZDT4's Test Results. ZDT4 is known to be a difficult multi-modal test problem. It has one true Pareto-optimal front and a huge number of local Pareto fronts (21⁹). PPREA is proving to be much more effective in solving this problem than NSGA-II in that, it is clearly outperforming NSGA-II (see Table 4).

From the 1st four figures in Fig. 3, one can see that the fronts returned by NSGA-II are clearly behind those returned by PPREA .

ZDT6's Test Results. Referring to Table 5, one can notice that even though the front returned by PPREA is smaller in size than that of NSGA-II, PPREA still outperforms NSGA-II in terms of taking bigger shares from the CPF, and of being the closest to the CPF as well. The last two figures in Fig. 3 shows some snapshots of the two fronts returned by the two contending MOEAs.

5 Concluding Remarks

A novel MOEA called PPREA (Potential Pareto Regions Evolutionary Algorithms) has been proposed based on a new fitness scheme and a new diversity technique.

NSGA-II of (Deb et al. 2001) has gained a worldwide popularity among MOEAs researchers. It is the most cited and the most used MOEA in the past couple of years. The new MOEA proposed in this work is compared against NSGA-II using the same test suite functions in which NSGA-II has excelled.

In some of the test results, PPREA clearly monopolised more of the Common Pareto Front (CPF) than NSGA-II did. In other test results, both algorithms *equitably* contributed to the CPF. However, the vector solutions returned by PPREA and left behind are much closer to the CPF than are those returned by NSGA-II. This result is anticipated since the very concept of PPREA is designed around the Expected-Improvement selection scheme which tend to drive the whole population towards the actual Pareto regions.

References

1. Deb K, Pratap, A., Agarwal, S., and Meyarivan, T. (2001). A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, vol 6(2), pp.182-197.
2. Burke, E.K. and Landa Silva, J.D. (2006). The influence of the Fitness Evaluation Method on the Performance of Multiobjective Optimisers, *European Journal of Operational Research*, Volume 169 issue 3, pp 875-897.
3. Burke, E.K., Landa Silva, J.D., and Soubeiga, E. (2005). Multi-objective Hyper-heuristic Approaches for Space Allocation and Timetabling, in: Ibaraki T., Nonobe K., Yagiura M. (eds.), *Meta-heuristics: Progress as Real Problem Solvers*, Springer, pp.129-158.
4. Hallam, N., (2005). State-of-the-art Multi-Objective Evolutionary Algorithms: Diversity Preservation and Archive Update Analysis, and Proposal of a New Evolutionary Algorithm, *PhD thesis submitted to the University of Nottingham*.
5. Hallam N., Blanchfield P., and Kendall G. (2005). Handling Diversity in Evolutionary Multiobjective Optimisation. In *Proceedings of 2005 IEEE Congress on Evolutionary Computation (CEC'05)*, pp. 2233-2240, Edinburgh, Scotland.
6. Knowles, J. D. and Corne, D. W. (1999). The Pareto archived evolution strategy: A new baseline algorithm for multi-objective optimization. *IEEE International Conference on Evolutionary Computation*, pp. 98-105.
7. Landa Silva, J.D. and Burke, E.K. (2004). Using Diversity to Guide the Search in Multi-objective Optimisation, In: Coello Coello C.A., Lamont G.B. (eds.), *Applications of Multi-Objective Evolutionary Algorithms, Advances in Natural Computation*, Vol. 1, World Scientific, pp. 727-751.
8. Zitzler, E. and Thiele, L. (1999). Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4), pp. 257-271.
9. Zitzler, E., Laumanns, M., and Thiele, L. (2001). SPEA2: Improving the strength pareto evolutionary algorithm. *Technical report*, Swiss Federal Institute of Technology.